

Priezvisko:

1 b	
2 b	
3 b	

Meno:

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	

Skúška trvá 70 minút.

Odpovede na otázky 1–12 vpíšte do tabuľky. Pri týchto otázkach sa hodnotia len odpovede v tabuľke (bez postupu). Odpoveď musí byť jednoznačná a čitateľná, inak má hodnotu 0 bodov.

V otázkach s ponúknutými odpoveďami je len jedna možnosť správna – do tabuľky vpíšte len písmeno, ktorým je označená odpoveď, ktorú vyberáte.

Odpoveď na otázku 13 píšete výlučne na list, na ktorom sa nachádza jej znenie.

Poškodený list nebude uznaný.

1. (1 b) Spracovanie výnimky v Jave sa realizuje

- (a) implementáciou metódy **try()** v triede výnimky
- (b) blokom kódu spracovania na mieste jej zachytenia
- (c) automaticky v metóde **main()**
- (d) implementáciou metódy **catch()** v triede výnimky
- (e) blokom kódu spracovania na mieste jej vyhodnenia

2. (1 b) Ak by ste v jazyku C++ implementovali sčítavanie a odčítavanie farieb, ktoré sú reprezentované inštanciami príslušnej triedy, na umožnenie zápisu týchto operácii pomocou operátorov + a – treba tieto operátory

- (a) preťažiť
- (b) virtualizovať
- (c) generalizovať
- (d) konkretizovať
- (e) prekonať

3. (1 b) V jazyku AspectJ

- (a) je možné zachytiť volanie metódy, ale nie aj nechať vykonať kód pred, po, a ani namiesto volania
- (b) je možné zachytiť volanie metódy a nechať vykonať kód pred, po alebo namiesto volania
- (c) je možné zachytiť volanie metódy a nechať vykonať kód iba pred alebo po volaní, ale nie aj namiesto volania
- (d) je možné zachytiť volanie metódy a nechať vykonať kód iba namiesto volania, ale nie aj pred alebo po volaní
- (e) nie je možné zachytiť volanie metódy

4. (1 b) Zahrnutie objektu jednej triedy v objekte inej triedy sa v objektovo-orientovanom programovaní označuje ako

- (a) integrácia
- (b) dedenie
- (c) segregácia
- (d) agregácia
- (e) generickosť

5. (1 b) Výhodou použitia aspektov v jazyku AspectJ je, že

- (a) automaticky opravujú chyby v programe
- (b) zrychľujú vykonávanie programu
- (c) umožňujú oddeliť prepletené záležitosti
- (d) umožňujú generovať lepšiu dokumentáciu než Javadoc
- (e) zvyšujú bezpečnosť

6. (1 b) Vlastnosti (properties) v jazyku C# umožňujú

- (a) zrýchliť vykonávanie programu
- (b) ovplyniť kód ako aspekty v jazyku AspectJ
- (c) vytvárať zástupcov funkcií
- (d) zachytiť udalostí
- (e) jednoduchšie zabezpečiť zapúzdrenie

7. (2 b) Daný je nasledujúci program v Jave:

```
class A extends Thread {
    C c;
    public A(C c) {
        this.c = c;
    }
    public synchronized void run() {
        for (int i = 0; i < 9999; i++)
            c.set0();
    }
}

class B extends Thread {
    C c;
    public B(C c) {
        this.c = c;
    }
    public void run() {
        for (int i = 0; i < 9999; i++)
            c.set1();
    }
}

public class C {
    private int d1 = 0, d2 = 0;

    public synchronized void set0() {
        if (d1 != d2)
            System.out.println(0);
        d1 = 0;
        d2 = 0;
    }

    public synchronized void set1() {
        if (d1 != d2)
            System.out.println(1);
        d1 = 1;
        d2 = 1;
    }

    public static void main(String[] args) {
        C c = new C();
        new A(c).start();
        new B(c).start();
    }
}
```

Výstupom tohto programu aj pri opakovanom vykonávaní

- (a) nebude nič alebo budú nuly a jednotky v nepravidelnom počte a striedaní
- (b) bude vždy desaťtisíc núl a jednotiek v nepravidelnom striedaní
- (c) bude vždy desaťtisíc jednotiek
- (d) nebude nič
- (e) bude vždy desaťtisíc núl

8. (2 b) Potrebné je prístupovať rovnakým spôsobom aj k entitám, ktoré sú jednoduché, aj k takým, ktoré sú zložené z iných entít. Ktorý návrhový vzor by ste použili?

- (a) Composite
- (b) Observer
- (c) Strategy
- (d) MVC
- (e) Visitor

9. (2 b) Čo všetko sa vypíše prostredníctvom príkazov `System.out.print()` po spustení nasledujúceho programu v Jave (po jeho úspešné alebo neúspešné ukončenie)?

```
class E extends Exception {}

public class M {
    public void m(char c) throws E {
        if (c == 'a')
            System.out.print("A");
        else
            throw new E();
    }

    public void f(char c) throws E {
        System.out.print("F");

        try {
            m(c);
        } catch (E e) {
            System.out.print("1");
        } finally {
            System.out.print("2");
        }
    }

    public static void main(String[] args) throws E {
        new M().f('a');
        new M().f('b');
        new M().f('a');
    }
}
```

10. (2 b) Vzhľadom na výsledný kód po kompilácii, pre príkazy **import** v Jave a **include** v C++ platí, že:

- (a) **include** nespôsobuje jeho nárast, kým **import** áno
- (b) **import** nespôsobuje jeho nárast, kým **include** áno
- (c) obidva príkazy za istých okolností môžu spôsobiť jeho nárast
- (d) ani jeden z nich nespôsobuje jeho nárast
- (e) **import** nespôsobuje jeho nárast, kým **include** len občas

11. (3 b) Trieda, ktorá reprezentuje vozidlo, poskytuje abstraktnú metódu **void** `umiestni(int miesto, Osoba osoba)`, ktorá umiestni danú osobu na dané miesto, ak je číslo miesta menšie alebo rovnaké ako počet miest vo vozidle, čo sa dá zistiť pomocou zodpovedajúcej metódy. Od tejto triedy sú odvodené triedy, ktorá reprezentujú dvojmiestne a štvormiestné vozidlo. Tieto triedy prekonávajú metódu `umiestni()` tak, že umiestnia danú osobu na dané miesto, ak je číslo miesta menšie alebo rovnaké ako počet miest vo vozidle.

Týmto sa dôsledky a predpoklady pôvodnej metódy zoslabujú, zosilňujú alebo nemenia? Je týmto dodržaný Liskovej princíp substitúcie (LSP)? Je v tomto prípade vhodné použitie dedenia?

Odpovedzte vo forme: *predpoklady / dôsledky / LSP / dedenie*. Položky *dôsledky* a *predpoklady* nahraďte jednou z možností *zoslabujú sa*, *zosilňujú sa* alebo *nemenia sa*. Položku *LSP* nahraďte jednou z možností *dodržený* alebo *nedodržený*. Položku *dedenie* nahraďte jednou z možností *áno* alebo *nie*.

12. (3 b) Čo sa vypíše po spustení nasledujúceho programu v Jave?

```
class A {
    public void x() {
        System.out.print("Ax");
    }
    public static void y() {
        System.out.print("Ay");
    }
}

class B extends A {
    public void x() {
        super.x();
        System.out.print("Bx");
    }
    public static void y() {
        A.y();
        System.out.print("By");
    }
}

class C extends B {
    public void x() {
        System.out.print("Cx");
    }
    public static void y() {
        System.out.print("Cy");
    }
}

public class M {
    public static void main(String[] args) {
        B o1 = new C();
        A o2 = new B();
        B o3 = new B();
        C o4 = new C();
        A o5 = new B();

        ((C) o1).x();
        ((C) o1).y();
        System.out.print(" ");

        o2.x();
        o2.y();
        System.out.print(" ");

        o3.x();
        o3.y();
        System.out.print(" ");

        ((A) o4).x();
        ((A) o4).y();
        System.out.print(" ");

        ((B) o5).x();
        ((B) o5).y();
    }
}
```

Priezvisko:

Meno:

13. (10 b) Postava v počítačovej hre môže u seba mať rôzne zbrane, ktoré môže použiť. Zbrane sa pri použití prejavajú, len ak postava má dostatok energie na ich aktiváciu. Na výstrel z pištole treba viac než 10% energie, kým na výstrel z luku treba viac než 50% energie. V hre v budúcnosti pribudnú ďalšie druhy zbraní, ktorých použitie tiež bude závisieť od energie postavy. Hra indikuje akcie prostredníctvom výpisov typu „výstrel z luku“ alebo „na výstrel z luku nestačí energia“.

Navrhnite a implementujte v Jave zodpovedajúce objektovo-orientované riešenie zohľadňujúce princípy objektovo-orientovaného programovania. Využite pritom najvhodnejší z návrhových vzorov Strategy, Observer, Visitor a Composite.

Základný návrh predložte vo forme náčrtu diagramu tried v UML, ktorý bude obsahovať najvýznamnejšie vzťahy, operácie a atribúty. Zoberťe pritom do úvahy návrhový vzor. V diagrame viditeľnosť prvkov nie je potrebné uvádzať.

V implementácii sa sústreďte na aplikačnú logiku – používateľské rozhranie nie je predmetom otázky.

Explicitne identifikujte prvky, ktorými sú modelované a implementované roly aplikovaného návrhového vzoru. Poskytnite príklad použitia, v ktorom vytvoríte príslušné objekty a spustíte ich interakciu.

Odpoveď bude hodnotená podľa nasledujúceho kľúča:

- zabezpečenie základnej funkčnosti – 4 b
- kvalita a flexibilita objektovo-orientovaného návrhu – 6 b

30

1 b

2 a

3 b

4 d

5 c

6 e

7 d

8 a

9 FA2F12FA2

10 b

11 nemenia sa / nemenia sa / dodržaný / áno

12 $Cx \setminus Cy \setminus Ax \setminus Bx \setminus Ay \setminus Ax \setminus Bx \setminus Ay \setminus By \setminus Cx \setminus Ay \setminus Ax \setminus Bx \setminus Ay \setminus By$ (akceptovaná je aj odpoveď bez medzier)

13 Vhodný je vzor Visitor. Postava je v role Element, pričom operácia `accept()` predstavuje aktiváciu zbrane. Zbrane sú v role Visitor, pričom ich skutočný prejav je daný operáciou `visit()`. Na pridanie ďalšej zbrane stačí od rozhrania alebo triedy, ktoré implementuje všeobecnú zbraň, odvodiť ďalšiu triedu a implementovať operáciu `visit()`.